

A faded, grayscale image of a statue of a man in historical attire, possibly a scholar or statesman, is visible on the left side of the slide. The statue is seated and appears to be holding a book or document.

# **Linux Info Tag**

## **Perl - Eine Free-Software-Kultur**

**-- Von helloworld bis OO --**

**Steffen Schwigon**  
**Dresden Perl Mongers**



## **Von helloworld bis OO**

# Hello world.

- hello.pl

```
print "Hello world.\n";
```

```
$ perl hello.pl  
Hello world.
```

- ◆ keine Includes o.ä.
- Kommandozeile

```
$ perl -e 'print "Hello world.\n"'  
Hello world.
```

# Shellersatz, 1

- listing.pl

```
#!/usr/bin/perl  
  
$ls = `ls *.png`;  
print "IMAGE FILES:\n" . $ls;
```

```
$ ./listing.pl  
IMAGE FILES:  
angel.png  
background4.png  
dummy.png
```

- ◆ Gesamte Ausgabe in String

# Shellersatz, 2

- doimg.pl

```
#!/usr/bin/perl

@ls = `ls *.png`;
foreach (@ls) {
    print "Mach was mit: $_";
}
```

```
$ ./doimg.pl
Mach was mit: angel.png
Mach was mit: background4.png
Mach was mit: dummy.png
```

- ◆ Array der Zeilen
- ◆ kontextabhängig
- ◆ Do The Right Thing (tm)

# use strict;

- doimgstrict.pl

```
#!/usr/bin/perl

use strict;
use warnings;

my @ls = `ls *.png`;
foreach (@ls) {
    print "Mach was mit: $_";
}
```

- saubererer Stil
- Verbote, Warnungen
- bevor wir größer werden

# Module

- DoImg.pm

```
package DoImg;  
  
sub mache {  
    my @ls = `ls *.png`;  
    foreach (@ls) {  
        print "Mach was mit: $_";  
    }  
}  
  
1;
```

- doimg2.pl

```
use lib ".";  
use DoImg;  
  
DoImg::mache();
```

# Klassen, 1

- DoImg2.pm

```
package DoImg2;

use Class::MethodMaker
    get_set      => [ 'result' ],
    new_hash_init => 'new';

sub mache {
    my $self = shift;
    my @ls   = `ls *.png`;
    foreach (@ls) {
        print "Mach was mit: $_";
    }
    $self->result (1);
}

1;
```

# Klassen, 2

- doimg++.pl

```
#!/usr/bin/perl  
  
use lib ".";  
use DoImg2;  
  
my $doimg = new DoImg2();  
$doimg->mache();
```

- Instanziierung
- Aufruf

# Distributionen, 1

- Bündel von Klassen/Modulen
- Standard-Verzeichnisstruktur
- *Build.PL*-File
- Standard-Distributionsfiles
  - ◆ *README*
  - ◆ *LICENSE*
  - ◆ *MANIFEST*
  - ◆ *Changes*
- Standard-Testumgebung
- Standard-Vorgehensweise

# Distributionen, 2

- Verzeichnisstruktur

```
./t
./lib
./lib/MyToolkit
./lib/MyToolkit/Db
./lib/MyToolkit/Gui
```

- Standardvorgehen

```
$ perl Build.PL           # (configure)
$ ./Build                 # make
$ ./Build test           # make check
# ./Build install        # make install
$ ./Build dist           # make dist
```

- große Komplexität handhabbar

# Schlußfolgerungen



# Schlußfolgerungen

- Perl ist für uns alle da

# Schlußfolgerungen

- Perl ist für uns alle da



- Danke.